DOCUMENT RESUME

ED 296 381 CS 506 201

AUTHOR Ku, Linlin
TITLE Copyright Protection for Computer Software: Is There a Need for More Protection?
PUB DATE Jul 88
NOTE 36p.; Paper presented at the Annual Meeting for the Association for Education in Journalism and Mass Communication (71st, Portland, OR, July 2-5, 1988).
PUB TYPE Speeches/Conference Papers (150) -- Information Analyses (070)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Software; *Copyrights; Court Litigation; Legal Problems; Mass Media Effects; Patents; Technological Advancement
IDENTIFIERS Contract Law; *Copyright Law 1976; *Copyright Law Amendments 1980; Software Copying; Software Piracy; Trade Secrets

ABSTRACT

Because the computer industry's expansion has been much faster than has the development of laws protecting computer software and since the practice of software piracy seems to be alive and well, the issue of whether existing laws can provide effective protection for software needs further discussion. Three bodies of law have been used to protect software: (1) trade secret and contract (effective protection if the software is a secret); (2) patent (the equivalent of a monopoly on a computer program); and (3) copyright. The Copyright Act of 1976 and the Computer Software Copyright Act of 1980 have both served to give some protection to computer software, but the issue of proving the copyright owner's rights have been infringed upon is still a problem. Several traditional and non-traditional tests have been developed to show substantial similarity between the copyrighted work and the pirated version, all of which are problematic. Two factors must be given attention when considering whether increased copyright protection should be given to software: any interpretation of the copyright statute should (1) not exceed principles suggested by legislative and judiciary history; and (2) not put too much emphasis on the copyright owner's economic interests, which will only impose a high cost on society. (One hundred and twenty-four notes are included.) (MS)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED296381

COPYRIGHT PROTECTION FOR COMPUTER SOFTWARE: IS THERE A NEED FOR MORE PROTECTION?

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Linlin Ku

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

A paper by Linlin Ku submitted in March 1988 to the Law Divison of the Association for Education in Journalism and Mass Communication.

Linlin Ku is a Doctoral Candidate in the Mass Media Ph.D. Program at Michigan State University, School of Journalism, East Lansing, MI 48824-1212, (517) 355-7989.

CS506201

BEST COPY AVAILABLE

COPYRIGHT PROTECTION FOR COMPUTER SOFTWARE: IS THERE A NEED FOR MORE PROTECTION?

INTRODUCTION

Because the computer industry's expansion has been much faster than has the development of laws protecting computer software, the issue whether existing laws can provide effective protection for software has been given much attention by scholars.[1] In 1980, the U. S. Congress amended the Copyright Act of 1976 which extends copyright protection to computer programs.[2] But the effectiveness of the amendment has been questioned.[3]

Courts have dealt with the issue of the copyrightability of computer programs. Several federal courts have declared that computer software, whether in source code or in object code, is copyrightable.[4] But there has not been any landmark case of the copyrightability of computer programs decided by the U. S. Supreme Court.

Software infringement suits may appear particularly troublesome to the courts. Since experienced computer programmers can copy the source code and hide evidence by translating it to another computer language, it will be very difficult for those jurors or judges who are unfamiliar with computers and software to detect similarities between two programs. So far at least two district courts have decided two software infringement cases by applying different tests.[5] That the practice of software piracy seems to be alive and well has resulted in a widespread argument that the nature of computer programs is quite different from that of traditional literary works and they deserve more legal protection.[6]

Can existing laws provide adequate protection? Is the 1980 amendment to the 1976 Copyright Act effective? Is there a need for more protection for software? What are the factors to be considered? This paper will review the legislative history of providing copyright protection for computer software and judiciary application of the copyright law. Trade secret and patent laws protecting software will also be briefly reviewed. The issues whether copyright can provide adequate protection for computer programs and whether there is a need for more protection will be discussed, and policies concerning providing proper protection will be proposed.

TYPES OF PROGRAMS

Source Code and Object Code

Computer programs are developed in stages. These stages usually include a definition of the program's task, a description, a listing of the program's steps and their expression in flow charts, the translations of these steps into statements. These statements are called the source code, often written in a high-level programming language such as FORTRAN and COBOL.

Source code statements must be translated into a binary form--computer machine language--before a computer can actually execute it. Machine language consists of only ones and zeros and is unintelligible even to highly skilled programmers and computer engineers. Source code may be translated, through intervention of a compiler or assembler program, into object code. In other words, object code is a compiled version of the source code, often physically embodied in punched cards, magnetic disks, magnetic tapes or silicon chips.

Operating Systems and Application Programs

Computer programs can also be distinguished into operating systems and application programs. A computer uses operating system programs internally to help execute application programs. Operating systems coordinate and control the flow of information among various parts of the computer, whereas application programs actually interact and perform a task for a user. Computer languages are considered as operating systems; application programs include word processing, spreadsheets, and so on.

LEGAL PROTECTION FOR COMPUTER SOFTWARE

Historically, three bodies of law have been used to protect software: trade secret and contract, patent and copyright. In this chapter, trade secret and contract and patent protections for computer programs will be briefly reviewed. Copyright protection will be discussed in later chapters.

Trade Secret and Contract Laws

The essence of trade secret[7] protection is nondisclosure.[8] The owner of a program may ask other people to sign a nondisclosure agreement, a contract that obligates the person signing it to keep the owner's secret. Thus, the owner has direct control over the program and any misappropriation is hampered.[9] Trade secret protection is often used by the owners of customized industrial or commercial computer programs.

Trade secret, in theory, is a highly effective means of software protection if the software is indeed a secret, and if the secrecy can be

maintained.[10] However, it may be difficult for the trade secret owner to prove that misappropriation has taken place.[11] Moreover, there is no federal trade secret law; each state may have different trade secret law. Finally, trade secret protection does not lend itself to protecting mass-marketed programs.[12] It is impractical to negotiate licenses with all potential users.

Patent Law

Patent protection is no more than the equivalent of a monopoly[13] on a computer program. If the software author or manufacturer owns a valid patent, he or she can prevent people from making, using and selling his or her program for seventeen years.[14] He or she can prevent people from using unauthorized copies.[15]

To qualify for patent protection, a computer program must meet three requirements by the Patent Act. First, it must fall within one of the four categories of the statutory subject matter: processes, machines, manufacture, or composition of matter.[16] Ideas,[17] laws of nature,[18] natural forces,[19] principles,[20] mathematical algorithms,[21] and mental processes[22] are specifically excluded from the subject matter. Next, it must be new and useful.[23] Third, it must be lack of obviousness. The program must be a true invention, not obvious to a skilled software writer.[24]

The debate regarding computer software patents has focused on whether a computer program or program-related invention can constitute statutory subject matter for a patent. This issue arose in 1972 in the first Supreme Court ruling on a program patent application, *Gottschalk v. Benson*.[25] Benson filed a patent application which claimed a method for converting binary-coded decimal numerals to binary numbers. The U. S.

Supreme Court decided that the method was a type of algorithm, a scientific truth that could not become a monopoly and concluded that the method could not be patented.[26]

The Court's next consideration of software patents came in the 1978 case of *Parker v. Flook*.[27] Flook applied for a patent on a method for updating alarm limits in catalytic conversion of hydrocarbons. The method involved the use of a mathematical algorithm to accomplish this process on a computer. The Court held that a process using a scientific principle is patentable only if the process itself is new and useful.[28] It stated that an algorithm could not be patentable because it had to be treated as part of the prior art and the patent, if issued, would be on a law of nature.[29]

The uncertainty of whether a mathematical formula or computer program is patentable led to the Supreme Court's consideration of *Diamond v. Diehr*.[30] The *Diehr* case involved a claim for a process for molding raw, uncured synthetic rubber into cured precision products. By a five-to-four majority, the Court held that the process was patentable subject matter.[31] The Court reiterated the *Benson* test by saying that "transformation and reduction of an article 'to a different state or thing' is the clue to the patentability of a process claim that does not include particular machines."[32] The Court said the methods for curing rubber involved just such a transformation and the use of a mathematical equation and programmed computer did not alter its decision.[33]

The *Diehr* decision is an important one since the Patent Office can no longer reject an application simply because the invention it describes involves a computer program.[34] A single computer program that is not connected with a specific statutory process will not be patentable, but an

inventor may use a computer in a process without worrying if its presence will jeopardize the patentability of an invention.[35] Because of this new development, it is possible that this area will be further modified in the coming years.[36]

Patents offer the highest degree of protection for intellectual property, but there are several disadvantages of patent protection for computer software. It is very time-consuming and expensive to get a patent.[37] By the time the patent issues, the program may be obsolete. Moreover, infringement of a patent on software will be difficult to detect and difficult to prove.[38] A third disadvantage is that patent protection and trade secrecy are incompatible. To get a patent, the software author must adequately make public how the program works,[39] and to do so will immediately terminate trade-secret protection.

COPYRIGHT LAW

In addition to trade secrets and patents, copyrights are a common form of protection for software. This chapter will review the legislative history of copyright protection.

Before the enactment of the Copyright Act of 1976, computer software had received limited copyright protection. Since 1964 the Copyright Office had been accepting computer programs for registration under the then current Copyright Act. This action was taken based upon the Copyright Office's "rule of doubt," and copyright protection was extended to computer programs that contained, if any, original literary expressions.[40] Mere

registration did not make them copyrightable, but did indicate the idea of software protection.

The Copyright Act of 1976

In order for a work to qualify for federal copyright protection, the Copyright Act of 1976 requires a work to be an original work of authorship "fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."[41] Following a non-exclusive list of seven categories of protected works,[42] the Act specifies that copyright protection does not extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery.[43] The copyright owner has the exclusive right to reproduce the work, prepare derivative works, distribute copies of the work and perform or display the work publicly.[44]

The Copyright Act of 1976 does not create a category to expressly protect computer programs. But the legislative history clearly suggested that Congress intended for computer programs to be eligible for copyright protection.[45] Computer programs fall well within the category of literary works.[46]

Congress had noticed the influence of the development of new technologies on the authorship, distribution and use of copyrighted works and the complexity of the problems before the enactment of the 1976 Copyright Act. In 1974, Congress created the National Commission on New Technological Uses of Copyrighted Works (CONTU) to evaluate the impact of new technologies on copyright laws.[47]

The 1978 CONTU Final Report

The CONTU submitted its Final Report on July 31, 1978.[48] It stated that "it was clearly the intent of Congress to include computer programs within the scope of copyrightable subject matter in the Act of 1976."[49] Because the commissioners could not agree what kinds of computer programs should be protected,[50] they suggested that the following definition of a computer program be added to section 101 of the Act without stating software is one of the categories of protected works: "A computer program is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."[51] They also recommended that a new section 117 be enacted to authorize the making of copies or adaptations of a program for backup and archival purposes.[52]

The Computer Software Copyright Act of 1980

At the end of 1980 the U. S. Congress passed the Computer Software Copyright Act,[53] adding a definition of computer program, a verbatim recommendation in the CONTU Final Report, to section 101. Section 117 was also amended to include verbatim the recommended statutory language proposed by CONTU to allow for copyrightability of computer programs. Section 117 states as follows:

> Notwithstanding the provisions of section 106, it is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
>
> (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or
>
> (2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that

> continued possession of the computer program should cease to be rightful.
>
> Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner.

Thus, computer programs are clearly recognized as copyrightable subject matter by virtue of the amended section 117. It should be noted that some programs may not be eligible for copyright protection if they either lack minimal originality or constitute the only way of accomplishing a specific result that they can be characterized as an idea rather than an expression of the idea.[54]

COPYRIGHT PROTECTION FOR OBJECT CODE

Although the 1980 amendment to the 1976 Copyright Act extends copyright protection to source code, the question of whether object code is a copyright subject matter had been widely discussed.[55] The dispute was derived from the confusion between object code and the medium used to store object code.[56] The judiciary history of interpreting copyright law and determining the copyrightability of object code reveals the complexity of the problem.

For many years a copyright could be used to protect only the written word. Then came the player piano. As this instrument became more popular, the competition between those who printed sheet music and those who made piano rolls got higher. Finally, the White-Smith Music Publishing Company charged the Apollo Company with stealing two of its copyrighted

songs and encoding them onto a piano roll.[57] The U. S. Supreme Court held a player piano roll was not a copy of the music it embodied but merely a part of the player piano mechanism, because both the roll and the music were imperceptible without the mechanism, the roll by itself was not meant for human communication, and did not constitute a copy under the then existing copyright statute.[58] Therefore, the principle was if the human eye could not read it, it was not copyrightable.

The 1909 Copyright Act, enacted shortly after the court's *White-Smith* decision, did not speak to this issue. For a time it seemed as though the *Apollo* ruling would be applied to software as well, since a perforated piano roll is not much different from a computer punch card.

Before the 1976 Copyright Act came into force, one important case addressing the issue of whether a program embodied in a ROM [59] could be a copy of a copyrighted program was *Data Cash Systems, Inc. v. JS & A Group, Inc.*[60] In 1977, Data Cash developed a chess game, "CompuChess." The essential program was encoded on a ROM. In 1978, JS & A began marketing a chess computer with a ROM identical to the CompuChess ROM. Data Cash brought a suit against JS & A for copyright infringement and unfair competition. An Illinois district court, relying on *White-Smith* decision, held that computer programs are copyrightable in their source code and flowchart phases, but the object code stored in a ROM was not a copy of the copyrightable computer source program because it was unintelligible to human beings and merely part of the computer hardware.[61]

The Seventh Circuit Court of Appeals affirmed the district court decision without reaching the question whether object code in a ROM was

copyrightable. The Seventh Circuit said the program could not be protected because it was published without copyright notice.[62]

The direct perception test of *White-Smith* was rejected by the 1976 Copyright Act, which broadened copyright protection to include anything tangible from which the author's work can be reproduced with the aid of a machine.[63] It was the intention of Congress that source code should be protected per se as literary works.[64] But the copyrightability of object code has been more problematic for the courts. The first case arising under the 1976 Act was *Tandy Corp. v. Personal Micro Computers, Inc.*.[65]

Tandy was the manufacturer of the Radio Shack TRS-80. It accused Personal Micro Computers of copying one program from TRS-80 and using it in its own computer designed for home use, the PMC-80. The program in question was installed in a ROM. The district court declared under the 1976 Copyright Act, "(1) a computer program is a work of authorship subject to copyright, and (2) that a silicon chip is 'tangible medium of expression,' within the meaning of that statute such as to make a program fixed in that form subject to the copyright laws."[66] The court concluded that unauthorized duplication of a silicon chip upon which a copyrighted program is embodied was not the use of a copyrighted program in conjunction with a computer, but simply the copying of a chip.[67]

In *Apple Computer, Inc. v. Franklin Computer Corp.*,[68] Apple sought a preliminary injunction to restrain Franklin from using, copying, selling, or infringing Apple's copyrights. The programs at issue were fourteen Apple's operating systems copied by Franklin to make ACE 100, Franklin's personal computer, Apple compatible. The motion was denied because the district court was not sure whether the creation of a computer program satisfied the

requirement of works of authorship, and whether an operating system program in binary code or one represented either in a ROM or by micro-switches was an expression.[69]

The district court's ruling was appealed to the Third Circuit Court of Appeals.[70] The Third Circuit decided several issues and reversed the earlier decision, granting a preliminary injunction. First, the court said although section 102(a) does not expressly list computer programs as works of authorship, the legislative history suggested that programs were considered copyrightable as literary works.[71] The court cited its decision in *Williams Electronics, Inc. v. Artic International, Inc.* [72] and held that "the copyrightability of computer programs is firmly established after the 1980 amendment to the Copyright Act."[73] The language of the 1976 Act and its legislative history suggested that it is not necessary to limit copyright to works designed to be read by human readers. The court concluded that a computer program, either in source code or object code, is a literary work and is protected from unauthorized copying.[74]

Second, the court cited its *Williams* decision that the embedding of object code in a ROM meets the statutory requirement of "fixation." It reaffirmed that a computer program in object code embedded in a ROM is a copyrightable subject matter.[75]

Third, Franklin conceded that it had copied the programs at issue in order to achieve complete compatibility, but argued that an operating system program was either a process, system, or method of operation and hence not copyrightable subject matter. The court stated that the statutory definition of a computer program makes no distinction between application programs and operating programs.[76] Franklin also argued that an operating

system program was too close to an idea to be copyrightable. The court found no evidence that the operating systems at issue represented the only means of expression of the idea underlying them and held that they are copyrightable.[77]

Finally, when deciding whether a preliminary injunction should be granted, the Third Circuit said Apple had made out a prima facie case of copyright infringement and was entitled to a preliminary injunction without a detailed showing of irreparable harm. It reasoned that Apple introduced substantial evidence of the considerable time and money it had invested in the development of the programs at issue. Franklin's wholesale copying would jeopardize Apple's investment and competitive position. Apple had met the requirement of irreparable harm needed to be granted a preliminary injunction.[78]

The Third Circuit affirmed a trend of providing copyright protection to object code, object code embedded in ROMs and operating system programs. Software authors should be happy with the *Franklin* decision. When seeking a preliminary injunction against unauthorized copying, they will have to show just a prima facie case or reasonable likelihood of success on the merits of their claims to raise a presumption of irreparable harm.

Although several courts have followed the legislative history and ruled that object code, regardless of its function, is copyrightable,[79] it is still unclear what types of programs may not be the subject matter of copyright and to what extent object code, object code embodied in a ROM, or operating systems can be protected by the copyright. Furthermore, the debate will not subside without a decision by the U. S. Supreme Court.

Another implication of the *Franklin* decision is its effects on the market. One scholar argued that allowing a manufacturer to obtain a copyright on an operating system program will severely restrict competition and can confer a de facto monopoly. He reasoned that a machine cannot function without using software. Software authors will write programs only for the best-selling machines. The only way for competing manufacturers to survive is to make their machines compatible with those best-selling machines. But true compatibility can be achieved only by using identical operating system programs. The decision may lead to domination of the hardware market by a few manufacturers.[80]

PROVING COPYRIGHT INFRINGEMENT

The issue of proving the copyright owner's rights have been infringed is more complicated. To prove the defendant has infringed upon the plaintiff's software copyright, the plaintiff must show ownership of a valid copyright on the program and copying by the defendant.[81] Ownership requires a copyrightable subject matter and proof of registration. A copyright registration certificate is prima facie proof of the validity of the copyright.[82] The next is to prove the defendant did in fact copy the program and the appropriation went so far as to be improper.[83]

Access

To prove the defendant has copied the copyrighted program, a plaintiff must show that the defendant had direct access to it, or show that the similarity between the two works is so striking that access can be

inferred.[84] For example, the plaintiff can show the defendant had a copy of the program in question, and the defendant was, or had access to the program through, a former employee of the plaintiff.[85]

Another common approach is to place nonfunctioning code in a program, and this code will remain even when the rest of the program is changed cosmetically. A similar way is to show the infringing program has the same bugs that are in the copyrighted program provided that a pirate usually makes superficial changes to conceal copying. The defendant will have to explain such coincidence. But the copyright owner will still have to show similarities between the two programs.[86]

Substantial Similarity

There must be a showing of substantial similarity between the two works to prove copyright infringement. Substantial similarity has been criticized for its subjective nature. The following three traditional tests may demonstrate this problem.

Line-by-line quantitative test

Lines of code are compared to identify lines of the two program instructions that are identical.[87] A major disadvantage of this quantitative test is that it is arbitrary--How much similarity must be shown to be substantial? Also, it is very easy for an experienced programmer to make some changes to avoid being identical.

Audience test

The audience test measures the effects of the infringing program upon the average reasonable man, who should be able to detect similarity between two programs without any aid or suggestion or critical analysis by

others.[88] Therefore, if one work appears similar to another in the eyes of the ordinary lay observer, it is likely to appear the same to most people. The audience test is invalid to evaluate similarity between two computer programs because an average reasonable man is unable to detect subtle software infringement without expert aid.

Extrinsic-intrinsic test

In *Sid & Marty Krofft Television Productions Inc. v. McDonald's Corp.*[89] the court fashioned a two-step analysis to grapple with the issue of substantial similarity. First, it should be considered whether a substantial similarity exists between the general ideas contained in two works--the extrinsic test. It should then be considered whether substantial similarity exists between the expressions of those ideas in the works--the intrinsic test.[90]

This test has two major flaws. First, since the 1976 Copyright Act does not protect ideas, the extrinsic test is irrelevant to a copyright infringement action. Second, the court held that dissection and expert testimony were irrelevant to the determination of substantial similarity in the expression, and the factfinder should consider only whether the accused work has captured the "total concept and feel" of the protected work.[91] Obviously, the extrinsic-intrinsic test is nothing but another audience test.

Nontraditional Tests

Because of the unique nature of software, several nontraditional tests have been fashioned to facilitate the determination of software infringement. These tests may be more appropriate for software infringement suits, but the problem has yet to be solved.

Audiovisual display test

To complement the comparison of program instructions, audiovisual sounds and screen displays can be compared to decide similarities between two programs. Screen displays resemble the traditional audiovisual works, and substantial similarities can be shown by finding similarities of sounds and screen displays of two programs.[92] One of the disadvantages of this test is that a pirate can copy a program and make a few changes to alter screen displays. Even when the results of two programs do not look alike, the programs may be substantially similar. Another disadvantage is that audiovisual test does not apply to some programs such as operating systems, which do not generate screen displays.

Course of development test

Because the traditional substantial similarity test has to rely on expert testimony which is hardly objective, it was suggested that course of development evidence be used, rather than evidence of substantial similarity, to prove copyright infringement.[93] The act of copying is the core copyright offense, and substantial similarity is no more than circumstantial evidence of copying. It was proposed that a presumption of copying should be raised if the plaintiff can convince the judge that the two programs are prima facie identical, virtually identical, or strikingly similar. If the plaintiff meets this burden, the burden will be shifted to the defendant to explain why the accused program is so similar to the protected work.[94]

A recent case[95] suggested the courts may be moving toward the point that course of development may be relevant to prove copyright infringement. As the first ruling to apply the derivative works[96] provision of copyright law to computer software, a Tennessee federal judge provided

some guidance on how software copyright holders could sustain a copyright infringement claim.[97] In this case SAS alleged that the S & H licensed a statistical program which operates only on IBM computers to obtain the source code, which it then used in an attempt to convert the copyrighted program to VAX computers. SAS claimed that the S & H product itself was either a copy of or a derivative work based on SAS, in either case making it an infringing work.

Judge Thomas A. Wiseman Jr. said that the definition of copy is focused on the "fixation" requirement, rather than on the degree of similarity shown to establish that a challenged work is a "copy."[98] He then determined whether the S & H product was substantially similar to SAS. Despite the fact that only forty-four of 186,000 lines of code were identical, he reasoned that whether similarities shown are of idea or expression is a question of fact; the substantiality of the similarity is also a question of fact, and the piracy of even a quantitatively small fragment may be qualitatively substantial.[99]

He then cited the definition of derivative work in the 1976 Copyright Act and said that the S & H product was well within the statutory definition of a derivative work, and not having been authorized by SAS, constituted an infringement of the copyright of SAS. S & H had engaged in improper conduct to achieve its conversion of SAS.[100]

The court's decision implied that what the defendant did while developing his or her program was more relevant to the question of infringement than whether his or her final product was substantially similar to plaintiff's. If courts inquire into the course of development of an allegedly infringing program, disassembling of a protected program may not be allowed, even if the new program is not substantially similar to the

copyrighted work. It was argued that copyright law should not prohibit disassembly of computer programs.[101] Furthermore, it prevents the copyright owner from exercising his rights of copying and adaptation specified in section 117 of the Act.[102]

Structural test

Another proposed test of program similarity is the structural test. It was argued that since copyright law recognizes writing computer instructions to implement subroutines and modules is an act of authorship, it should also recognize organizing and connecting the modules and subroutines as an act of authorship.[103]

One district court went beyond measuring the amount of literal identity between program instructions and considered structural similarity.[104] Jaslow Dental Lab was accused of obtaining a copy of the source code for a dental lab office management program "surreptitiously and without consent." Jaslow recast the program in a different programming language. No line of code between the two programs was identical. But the court found similarity between the two programs by defining "the 'expression of the idea' in a program as the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating and producing useful information either on a screen, print-out or by audio communication."[105] This definition, one scholar argued, does not refer to the external functions of a program in response to the user; it must be the internal functions that the program performs on data.[106] While external functions are utilitarian functions or ideas, a showing of similarity between the internal functions of two programs should establish copyright infringement. Under the structural

test, the plaintiff will not be subject to a measure of quantitatively substantial similarity, and the literary work copyright can provide more effective protection for software.[107]

However, the structural test has its own shortcomings. It is unclear whether elements of structural design are copyrightable. Furthermore, the concept of "structure" is rather vague. The court's description of what Jaslow appropriated was so broad as to include ideas, while copyright protects the expression of an idea, not the idea itself.

In short, all of the above similarity tests have their disadvantages. Courts and juries are likely to flounder in a sea of contradictory expert testimony about whether a particular program actually infringes another. Obviously, courts should develop effective methods to determine copyright infringement.

POLICY CONSIDERATIONS

Due to the development of new technology, whether existing laws can effectively protect intellectual properties to encourage the progress of arts and science has been questioned. This chapter will focus on policies of copyright protection for software. The effectiveness of the 1980 Computer Software Copyright Act will be assessed.

Economic Force

One of the basic considerations is that computer software should be protected. The basis for this consideration was clearly stated in the CONTU economic report.[108] It stated that to encourage the creation and

dissemination of computer programs, authors must be given some form of legal protection. It found that the computer software industry has high development costs but low marginal costs. The cost of producing the first program is high while the cost of duplicating that program for future sales is relatively low. Software pirates could obtain significant competitive advantage by purchasing a developer's program on the open market, reproducing it for general distribution, and then selling copies in direct competition with the developer at drastically lower prices. Consequently, computer programs require legal protection.[109]

Clearly, the predominant economic rationale for copyright protection is that literary, artistic, and scientific works are knowledge goods that can be appropriated. To encourage the production of creative works, it is necessary to provide their creators with economic incentives.[110]

Furthermore, new technologies have facilitated unauthorized use of copyrighted works, making unauthorized duplication easy and inexpensive. Impacts of technologies on the copyright owner has resulted in a strong argument about protecting the copyright owner's economic rights. Two scholars argued that copyright protection for computer software should be interpreted as a function of economic imbalances. They stated that any interpretation of the 1980 amendment should focus on the economic imbalance and unfairness that would result if computer programs were not protected by copyright, and that any determination of copyright infringement should focus on the economics of copying.[111]

Although the copyright owner's economic interests should be carefully redressed, there is always a counterargument: copyrights limit the free flow of information and the public's right to use information.[112] The conflict

between the copyright owner's need for protecting economic rights and the public interest in the free flow of information is inevitable that Congress should balance the competing interests between the two by carefully examining how copyright policies should respond to changes in an information society. Congress should provide a"neutral playing field" for competitors in the computer software market to prevent dominance by a few software manufacturers.

The Appropriateness of Copyright

After reviewing existing laws protecting software, the CONTU found that copyright is the most proper mechanism to protect computer programs:

1) Copyright is applicable to virtually all programs and is well suited to mass-marketed programs;

2) Copyrights are more easily enforced than patents, and

3) Copyrights give protection to the owner without denying the opportunity for others to use and benefit from the owner's work.[113]

Copyrights are the most appropriate legal protection available for computer software when compared with other legal means such as patents and trade secrets. The debate centered around the appropriateness of copyright protection is that only some types of programs are copyrightable. Since the 1976 Copyright Act was amended to include computer programs, it is up to the courts to interpret the 1980 amendment.

The Effectiveness of the 1980 Computer Software Copyright Act

The CONTU commissioners agreed that software should be protected and copyright is the proper mechanism to protect software, and

recommended the copyright statute be amended to extend copyright protection to software.[114] The recommended definition of a computer program was included in the 1980 amendment, which indicated a functional approach. Whether object code is copyrightable depends on the interpretation of "statements or instructions," "to be used," "directly," "indirectly" and "a certain result." It is suggested that the interpretation of any of these terms be based not only on the very nature of a computer program but also on legislative history.

One explanation was that this definition implies the copyrightability of object code which is used "directly" in a computer, as opposed to source code which is used "indirectly" in a computer.[115] However, this did not clarify what types of programs should be protected. A decision should be made by the U. S. Supreme Court that certain forms of software are not proper subject matter of copyright.

In addition, section 117 was amended to provide for the "fair use" of computer programs in the preparation of "copies or adaptations." Legitimate uses include inputting the program into a computer, running the program and archiving the program.

One of the problems of this section is what constitutes an "adaptation."[116] According to CONTU's description, the adaptation right allows the program owner the right to make changes necessary to enable the use for which it was both sold and purchased. This right includes the conversion of a program from one higher-level language to another to facilitate use and the right to add features to the program that were not available at the time of acquisition.[117] While the CONTU indicated that the adaptation right "could only be exercised so long as it did not harm the

interests of the copyright proprietor,"[118] this broad protection of the copyright owner's interests was not reflected in the language of the statute. Therefore, the interpretation of what constitutes an adaptation is left to the court, which is decisive in determining software infringement.

Another problem with section 117 is the lack of guidance regarding what constitutes a fair use of copies or of adaptations of a program.[119] Again, the interpretation of whether a program is used to prepare backup copies and adaptations is a fair use is left to the court.

It was also argued that although the 1980 amendment provides a legal doctrine to protect against infringement, it does not provide any practical protection from widespread program piracy. One scholar proposed harsher penalties for willful copyright infringement of programs and compulsory licenses as requirement for making and distributing software on the ground that software copyright owner's economic interests should be well protected.[120]

Some Suggestions

Seeking more protection

Before the uncertainty of copyright protection for computer software can be clarified, software authors may want to combine several means to obtain the highest possible protection they can. It is suggested that software authors obtain additional protection for programs that generate screen displays by claiming an audiovisual work copyright in the expressive elements of a program's screen display. For customized industrial or commercial programs the software author may combine trade secrecy, compulsory licensing, and copyright. For mass-marketed programs the

software author may have to rely on copyrights with additional schemes such as price discrimination and recovery of payments when programs are copied.

Judicial interpretations

Although the 1976 Copyright Act extends protection to software, courts had been confused between the protected form of expression and the medium of expression until the *Franklin* decision[121] that eliminated the distinctions among types of programs, forms of expression and different mediums containing the program. The Third Circuit Court of Appeals affirmed that software is protectable under copyright law. The only limitation the court expressed was that operating systems would not be protectable if it could be shown that the function of the operating systems could be accomplished by only one form of expression of the software.[122] It is suggested that courts develop meaningful guidelines for distinguishing copyrightable ROM-embedded object code from uncopyrightable utilitarian uses of ROM-embedded hardware.

To determine software copyright infringement cases, courts should develop effective expert testimony to decide whether the accused program is substantially similar to the copyrighted program, and this can be very technical. While the traditional audience test may no longer relevant in deciding similarities between two programs, the course of development test may go so far as to decide two programs are similar without finding any identical line of code between the two programs. A proponent for the structural test claimed it can balance the traditional similarity and the course of development tests.[123] But the structural test is rather vague and can provide no better guidelines for courts to follow than can the traditional

similarity test. Since there seems no single solution to this problem, more than one method of analysis may be required to determine similarities between two programs. For example, the quantitative test may be complemented by the audiovisual display test. Any proposal for effective expert testimony requires guidelines for finding substantial similarities between the two programs.

CONCLUSION

It is undisputed that software should be protected to encourage the creation and dissemination of computer programs. However, whether more than existing copyright protection should be given to software needs more consideration. For one thing, any interpretation of the copyright statute should not exceed principles suggested by legislative and judiciary history. For another, any interpretation of the copyright statute should not put too much emphasis on the copyright owner's economic interests, which will only impose a high cost on society. The interest of the public should be carefully addressed. Whether a program results in useful work or copyrightable expression, the nature of the program and the author's expression in it must be carefully considered to prevent copyright from restricting the techniques and methods of programming itself. After all, the purpose of the copyright statute is to "promote the progress of science and useful arts, by securing for limited times to authors...the exclusive right to their perspective writings...."[124] Only by carefully balancing the interest of the public and the interest of copyright owners will we be able to promote computer software as a new useful art and science.

[1] See Susan A. Dunn, "Defining the Scope of Copyright Protection for Computer Software," *Stanford Law Review* 38 (1986): 497-534; Bruce Perelman, "Proving Copyright Infringement of Computer Software: An Analytical Framework," *Loyola of L. A. Law Review* 18 (1985): 919-48; Andrew G. Rodau, "Protecting Computer Software: After Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d. Cir. 1983), Does Copyright Provide the Best Protection?" *Temple Law Quarterly* 57 (1984): 527-52; J. Clark Kelso and Alexandra Rebay, "Problems of Interpretation Under the 1980 Computer Amendment," *Santa Clara Law Review* 23 (1983): 1001-35; and Micky T. Mihm, "Software Piracy and the Personal Computer: Is the 1980 Software Copyright Act Effective?" *Computer Law Journal* IV (1983): 171-93.

[2] Copyright Act of 1976, Pub. L. No. 96-517, 94 Stat. 3028 (codified as amended at 17 U.S.C. SS 101, 117 (1982)).

[3] See Mihm, supra, Note 1.

[4] See Apple Computer, Inc. v. Formula International Inc., 725 F.2d 521 (9th Cir. 1984); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3rd Cir. 1983), cert. dismissed, 104 S. Ct. 690 (1984); and Williams Electronics, Inc. v. Artic International, Inc., 685 F.2d 870 (3rd Cir. 1982).

[5] In SAS Institute, Inc. v. S & H Computer Systems, Inc. , 605 F. Supp. 816 (M.D. Tennessee 1985), the court applied the derivative works provision of copyright law to computer software, while in Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc. , 609 F. Supp. 1307 (D.C. Pa. 1985), the court relied on a test of structural similarity to decide whether the two programs at issue were similar.

[6] See *Final Report of the National Commission on New Technological Uses of Copyrighted Works*, (Washington, D. C.: Library of Congress, 1979). p.12.

[7] A trade secret can be defined as "any formula, pattern, device or compilation of information which is used in one's business, and which gives him an opportunity to obtain an advantage over competitors who do not know or use it." See Restatement of Torts §757 comment b (1939).

[8] Restatement §757.

[9] Ibid.

[10] Frederic William Neitzke, *A Software Law Primer* (New York: Van Nostrand Reinhold Company, 1984), p. 35.

[11] Michael C. Gemignani, *Computer Law* (Rochester, N.Y.: The Lawyers Co-operative Publishing Co., 1985), pp. 652-654.

[12] As the personal computer industry has been growing rapidly, protecting mass-marketed programs has become a major concern of the software developers.

[13] See Valmont Industries, Inc. v. Yuma Mfg. Co., 296 F. Supp. 1291, 1294 (D. C. Colo. 1969).

[14] 35 U. S. C. A. §154.

[15] 35 U. S. C. A. §281.

[16] 35 U. S. C. A. §101.

[17] Gottschalk v. Benson, 409 U.S. 63, at 67, 93 S. Ct. 253, 34 L. Ed. 2d 273 (1972).

[18] Application of Christensen, 478 F.2d 1392 (C. C. P. A. 1973).

[19] Ibid.

[20] Ibid.

[21] Supra, Note 17, at 67-68.

[22] Ibid., at 67.

[23] Supra, Note 16.

[24] 35 U.S.C.A. § 103.

[25] Supra, Note 17.

[26] Ibid., at 71-72.

[27] 437 U.S. 584, 98 S. Ct. 2522, 57 L. Ed. 2d 451 (1978).

[28] Ibid., 98 S. Ct., at 2526.

[29] Ibid., at 2523-2524.

[30] 450 U.S. 175, 101 S. Ct. 1048, 67 L. Ed. 2d 155 (1981).

[31] Ibid.

[32] Ibid., 101 S. Ct. at 1055.

[33] Ibid.

[34] See Gemignani, Note 11, p. 628.

[35] See Arthur R. Miller and Michael H. Davis, *Intellectual Property: Patents, Trademarks, and Copyright* (St. Paul, Minn.: West Publishing Co., 1983), p. 34.

[36] The Supreme Court has asked Congress to settle the issue of the patentability of computer software, but Congress has not dealt with this issue. See Gottschalk v. Benson, supra, Note 17, and Parker v. Flook, supra, Note 27.

[37] It may take as long as three years to obtain a patent, and the cost may be as high as $100,000. See Rodau, supra, Note 1, 532.

[38] Ibid.

[39] 35 U. S. C. A. §112.

[40] See Howard B. Abrams, "Current Developments in Copyright Law: The Subject Matter Issues," in *Current Developments in Copyright Law 1985* (New York: Practising Law Institute, 1985), pp. 53-54.

41 17 U.S.C.A. § 102(a).

42 The seven categories include literary; musical; dramatic; pantomimes and choreographic; pictorial, graphic and sculptural; motion pictures and other audiovisual works; and sound recordings. Ibid., § 102.

43 Ibid., § 102(b).

44 Ibid., § 106.

45 See Abrams, supra, Note 40, pp. 54-55.

46 Ibid.

47 See Robert W. Frase, "National Commission on New Technological Uses of Copyrighted Works (CONTU)," in *Modern Copyright Fundamentals*, eds. Ben H. Weil and Barbara Friedman Polansky (New York: Van Nostrand Reinhold Company, 1985), p. 255.

48 Supra, Note 6.

49 Ibid., p. 16.

50 The commission recommended that the 1976 Copyright Act be amended to include computer programs as proper subject matter of copyright. In his dissenting opinion, commissioner John Hersey said the act of 1976 should be amended to make it explicit that copyright protection does not extend to a program that is used to control computer operations. See Frase, supra, Note 47, pp. 256-257.

51 Supra, Note 6, p. 22.

52 Ibid.

53 Public Law 96-517, 94 Stat. 3028 (December 12, 1980).

54 17 U. S. C. A. §102(a).

55 See Abrams, supra, Note 40, pp. 56-65; Allen R. Grogan and Michael J. Kump, "The Broader Meanings of Apple v. Franklin in the Development of Compatible Operating Systems and in Determining Standards for Injunctive

Relief," in *Modern Copyright Fundamentals*, supra, Note 47, pp. 302-308; and Leslie Wharton, "Use and Expression: The Scope of Copyright Protection for Computer Programs," *Computer Law Journal* VI (1985): 433-468.

[56] See CONTU, *Final Report*, supra, Note 6.

[57] See White-Smith Music Publishing Co. v. Apollo Co., 209 U.S. 1, 28 S. Ct. 319, 52 L. Ed. 655 (1908).

[58] Ibid., 209 U. S., at 17-18.

[59] ROM stands for "read only memory." ROM is fixed, and cannot be easily modified. ROM is used to store operating system programs, compilers, and communication interfaces. It usually involves a chip that contains circuitry embodying the instructions built into it by the designer.

[60] 480 F. Supp. 1063 (N.D. Ill. 1979), 628 F.2d 1038 (7th Cir. 1980).

[61] 480 F. Supp. 1063 (N.D. Ill. 1979), at 1069.

[62] 628 F.2d 1038 (7th Cir. 1980), at 1041.

[63] Supra, Note 41.

[64] In the House Report from the 94th Congress, it is stated that literary works include computer programs "to the extent that they incorporate authorship in a programmer's expression of original ideas, as distinguished from the ideas themselves." See H. R. Rep. No. 1476, 94th Cong., 2d Sess., 54 (1976).

[65] 524 F. Supp. 171 (N.D. Cal. 1981).

[66] Ibid., at 173.

[67] Ibid., 175.

[68] 545 F. Supp. 812 (E.D. Pa. 1982).

[69] Ibid., at 820-821.

[70] 714 F.2d 1240 (1983), cert. dismissed, 104 S. Ct. 690 (1984).

[71] 714 F.2d, at 1247.

[72] 685 F.2d 870 (3rd Cir. 1982).

[73] Ibid., at 875.

[74] Supra, Note 71, at 1248-1249.

[75] Ibid., at 1249.

[76] Ibid., at 1251-1252.

[77] Ibid., at 1252-1253.

[78] Ibid., at 1254.

[79] Supra, Note 4.

[80] Gemignani, supra, Note 11, pp. 598-599.

[81] 3 M. Nimmer, Nimmer on Copyright §13.01 (1984).

[82] 17 U. S. C. A. §410(c).

[83] See Arnstein v. Porter, 154 F.2d 464 (2nd Cir. 1946).

[84] Ibid., 468.

[85] Daniel Remer, *Legal Care for Your Software* (Berkeley, Calif.: A Nolo Press Book, 1984), p. 159.

[86] Ibid., p. 160.

[87] See Midway Mfg. Co. v. Strohon, 564 F. Supp. 741 (N.D. Ill. 1983).

[88] See Harold Lloyd Corp. v. Witwer, 65 F.2d 1 (9th Cir. 1933).

[89] 562 F.2d 1157 (9th Cir. 1977).

[90] Ibid.

[91] Ibid., at 1167.

[92] Perelman, supra, Note 1, 945-946.

[93] Ibid., 942-945. Also see Kelso and Rebay, supra, Note 1, 1028.

[94] Ibid.

[95] SAS Institute, Inc. v. S & H Computer Systems, Inc., 605 F. Supp. 816 (M.D. Tenn.1985).

[96] A derivative work is defined as work "based upon one or more pre-existing works, such as a translation, ..., abridgement, condensation, or any other form in which a work may be re-cast, transformed or adapted." 17 U.S.C.A. §101.

[97] Supra, Note 95.

[98] Ibid., at 829.

[99] Ibid., at 829-830.

[100] Ibid., at 831.

[101] Dunn, supra, Note 1, 517-518.

[102] Ibid., 518-520.

[103] Ibid., 526.

[104] See Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 609 F. Supp. 1307 (E.D. Pa. 1985).

[105] Ibid., at 1320.

[106] Dunn, supra, Note 1, 528.

[107] Ibid., 529.

[108] See *National Commission on Technological Uses of Copyrighted Works, Economics of Property Rights as Applied to Computer Software and Data Bases* (1977).

[109] CONTU *Final Report*, supra, Note 6, 11, and CONTU *Economics of Property Rights as Applied to Computer Software and Data Bases*.

[110] See Robert P. Benko, *Protecting Intellectual Property Rights* (Washington, D. C., American Enterprise Institute for Public Policy Research, 1987), p. 21.

[111] See Kelso and Rebay, supra, Note 1, 1002-1003, 1035.

[112] David Ladd, "The Future of Copyright," in *Modern Copyright Fundamentals*, supra, Note 47, pp. 442-443.

[113] Supra, Note 6, p. 19.

[114] Supra, Note 6.

[115] Kelso and Rebay, supra, Note 1, 1020.

[116] Ibid., 1029.

[117] Supra, Note 6, p. 13.

[118] Ibid.

[119] Kelso and Rebay, supra, Note 1, 1030.

[120] Mihm, supra, Note 1, 189.

[121] Apple v. Franklin, supra, Note 4.

[122] Ibid.

[123] Dunn, supra, Note 1, 529.

[124] U. S. Const. art. I, § 8, cl. 8.